Software Metrics

Kristian Sandahl

LINKÖPING UNIVERSITY

Agenda:

Terminology and scope

Quality factors revisited

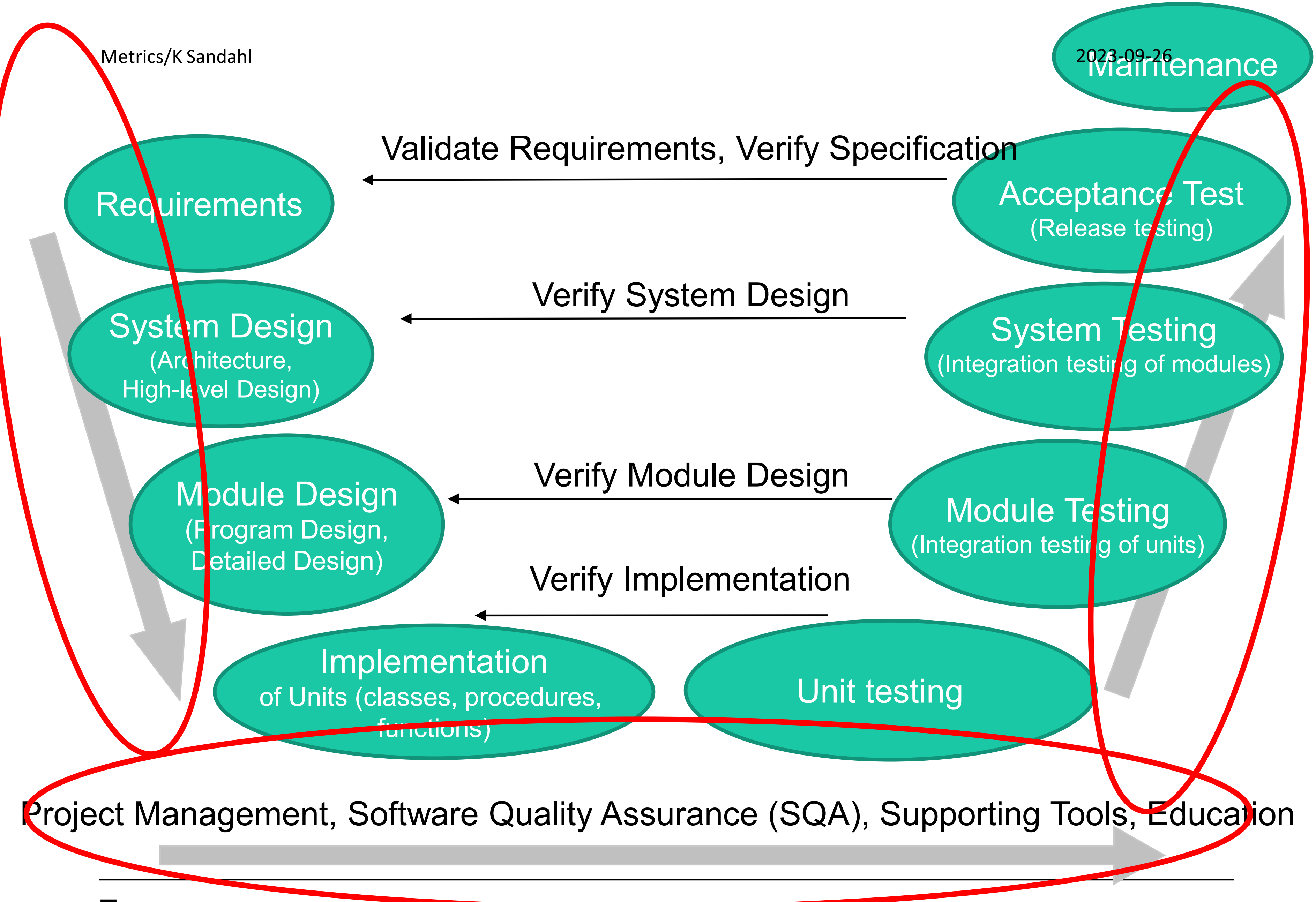
Model of failure-based metrics

Usability metrics

Cyclomatic complexity

Examples of metrics and quality factors

Practical use


Maintenance
Validate Requirements, Verify Specification
Requirements
Acceptance Test
(Release testing)
Verify System Design
System Design
(Architecture,
High-level Design)
System Testing
(Integration testing of modules)
Verify Module Design
Module Design
(Program Design,
Detailed Design)
Module Testing
(Integration testing of units)
Verify Implementation
Implementation
of Units (classes, procedures,
functions)
Unit testing
Project Management, Software Quality Assurance (SQA), Supporting Tools, Education

Measurement - metrics

Most common use:

- Measurement – directly measured on:
  - Document, no of pages
  - Design, no of model elements
  - Code, no of lines
  - Process, iteration length
  - Quality, avg no of hours to learn a system
- Metrics – is a combination of measurements, e.g. number of faults found in test/hours of testing

Agenda:

~~Terminology and scope~~

Quality factors revisited

Model of failure-based metrics

Usability metrics

Cyclomatic complexity

Examples of metrics and quality factors

Practical use

ISO/IEC 25010 (2011)

"The quality of a system is the degree to which the system satisfies the stated and implied needs of its various stakeholders, and thus provides value. Those stakeholders' needs (functionality, performance, security, maintainability, etc.) are precisely what is represented in the quality model, which categorizes the product quality into characteristics and sub-characteristics" (www.iso2500.com)

Sometimes called quality factors

Today's examples

Measuring these requires both research, experience and imagination.

NB: Coupling and cohesion

- Coupling and cohesion are contributing to the different quality factors
- There are many ways of measuring coupling and cohesion
- In contrast, the system used in labs in TDDC88/725G64 mention coupling and cohesion as metrics, **that is insufficient**

Agenda:

~~Terminology and scope~~

~~Quality factors revisited~~

Model of failure-based metrics

Usability metrics

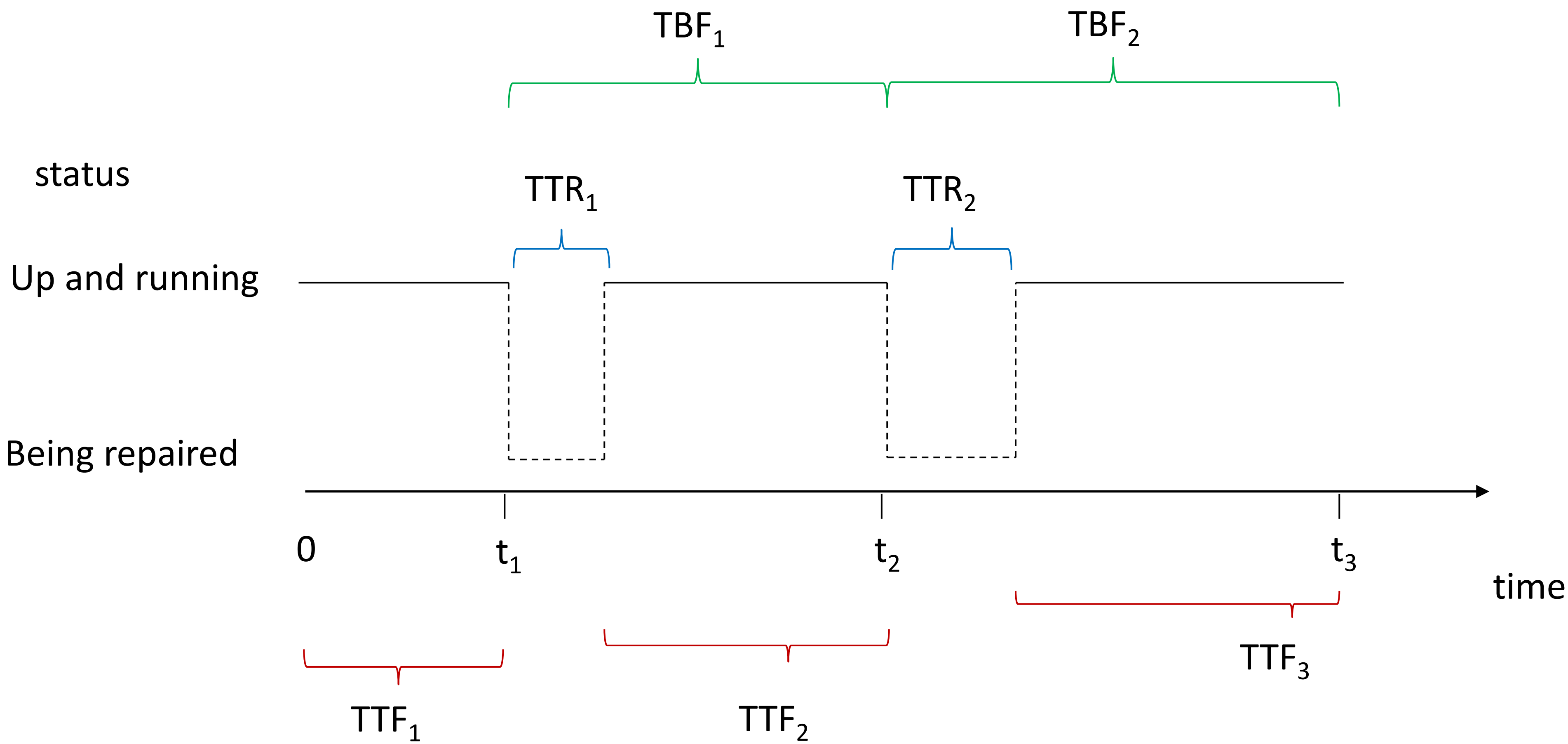
Cyclomatic complexity

Examples of metrics and quality factors

Practical use

Course standard

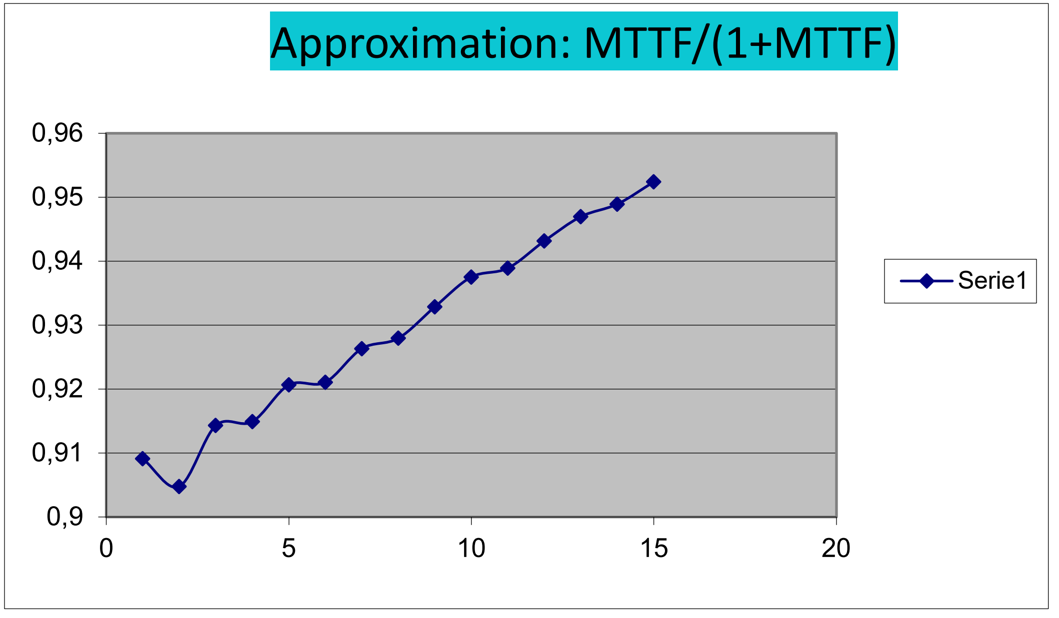
Simplified model with repair time



Reliability growth model

The probability that the software executes
with no failures during a specified time interval

Failure number	Failure time
1	10
2	19
3	32
4	43
5	58
6	70
7	88
8	103
9	125
10	150
11	169
12	199
13	232
14	260
15	300

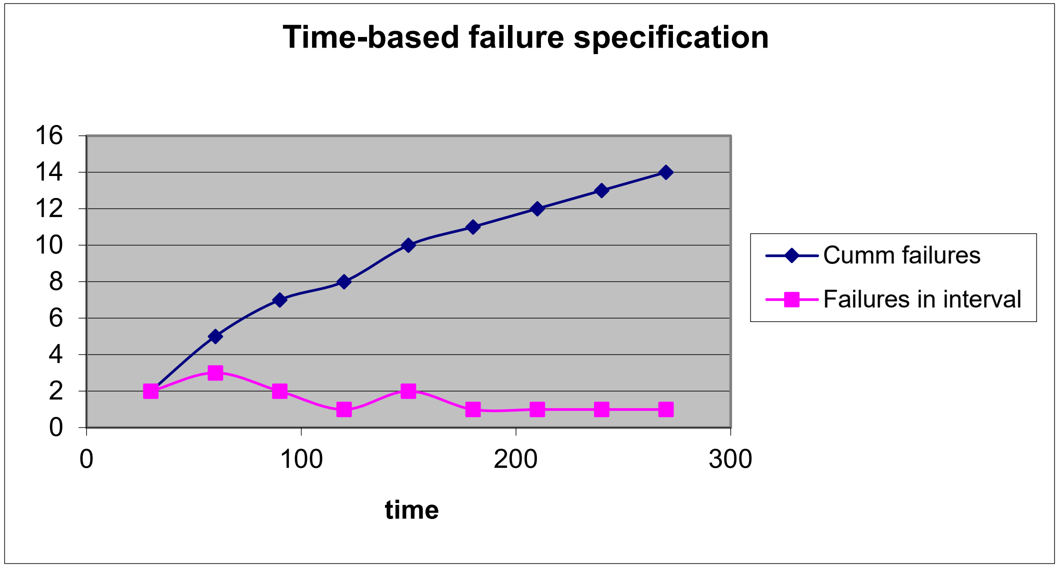


Easier to manage: Failure intensity

Default: [failures / hours of execution time]

Or other natural unit

Time	Cumm failures	Failures in interval
30	2	2
60	5	3
90	7	2
120	8	1
150	10	2
180	11	1
210	12	1
240	13	1
270	14	1
300	15	1



Another approximation: $\lambda = (1-R)/t$

Similar pattern: Availability and Maintainability

- Measure Mean Time To Repair (MTTR) and Mean Time To Failure (MTTF)
- Availability, A:
- A = MTTF/(MTTF+MTTR)

- Measure Mean Time To Repair (MTTR)
- Maintainability, M:
- M = 1/(1 + MTTR)

Agenda:

~~Terminology and scope~~

~~Quality factors revisited~~

~~Model of failure-based metrics~~

Usability metrics

Cyclomatic complexity

Examples of metrics and quality factors

Practical use

Measure usability?

Relevance	Efficiency	Learnability	Attitude
• number of good and bad features recalled by users • number of available commands not invoked by users • number of available commands invoked by users • number of times user needs to work around a problem • percent of task completed	• time to complete a task • percent of task completed • percent of task completed per unit time (speed metric) • time spent in errors • number of commands used • frequency of help and documentation use • time spent using help or documentation	• ratio of successes to failures (over time) • time spent in errors • percent or number of errors • number of commands used • frequency of help and documentation use • time spent using help or documentation • number of repetitions of failed commands	• percent of favorable/unfavorable user comments • number of good and bad features recalled by users • number of users preferring the system • number of times user loses control of the system • number of times the user is disrupted from a work task

Agenda:

~~Terminology and scope~~

~~Quality factors revisited~~

~~Model of failure-based metrics~~

~~Usability metrics~~

Cyclomatic complexity

Examples of metrics and quality factors

Practical use

Computation of cyclomatic complexity

Cyclomatic complexity has a foundation in graph theory and is computed in the following ways:

1. Cyclomatic complexity V(G), for a flow graph, G, is defined as:

   $V(G) = E - N + 2P$

   E: number of edges

   N: number of nodes

   P: number of disconnected parts of the graph

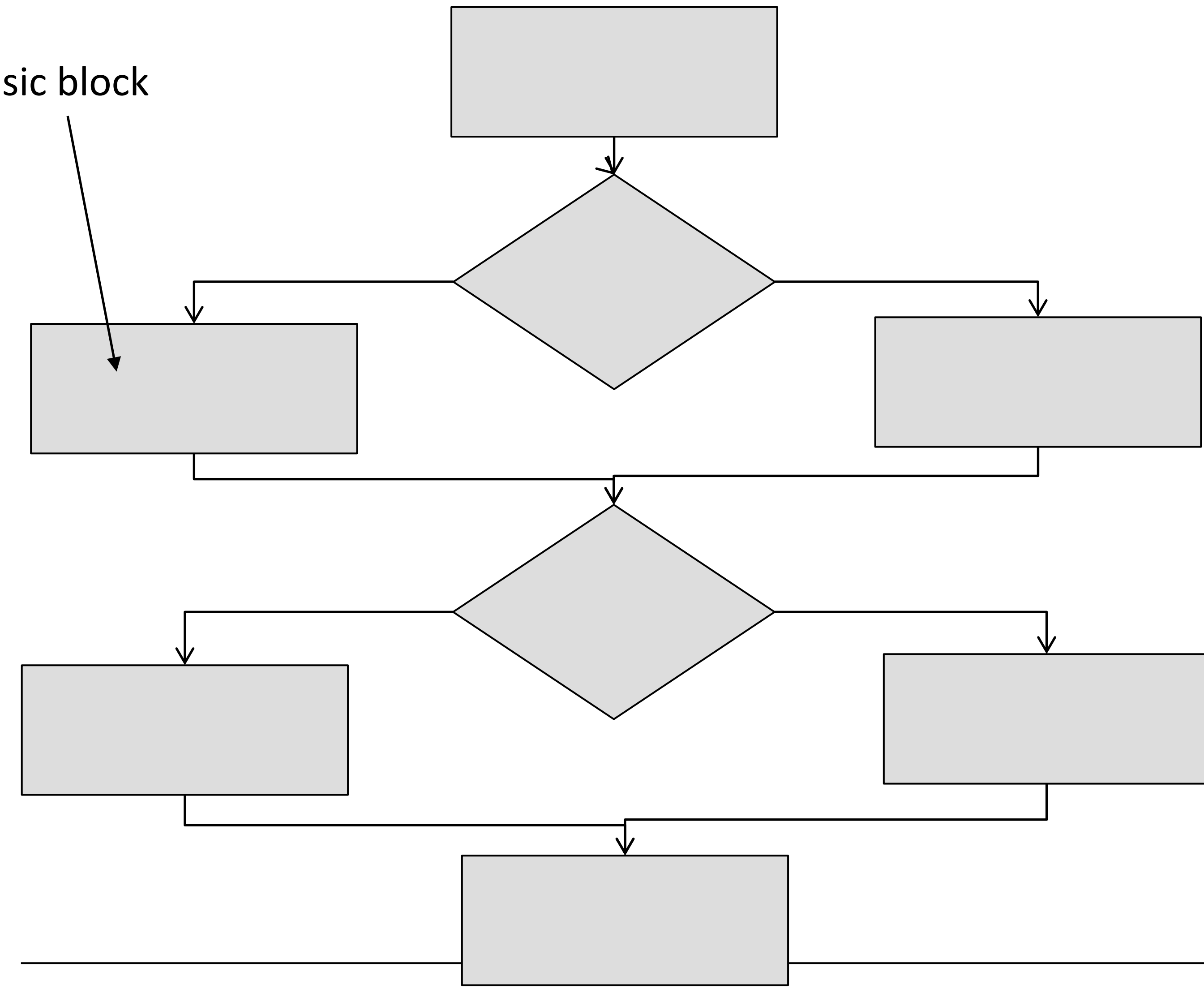
2. Cyclomatic complexity V(G), for a flow graph, G, with only binary decisions, is defined as:

   $V(G) = b + 1$

   b: number of binary decisions

Control-flow

Basic block



$E = 9$

$N = 8$

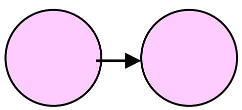
$P = 1$

$V = 3$

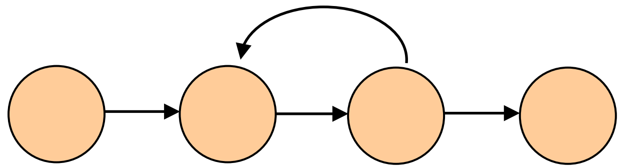
B = number of binary decision points

$V = B+1$

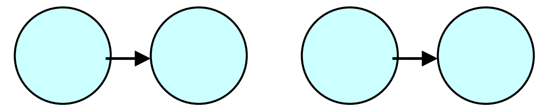
Examples of Graphs and calculation of McCabe's Complexity Metric



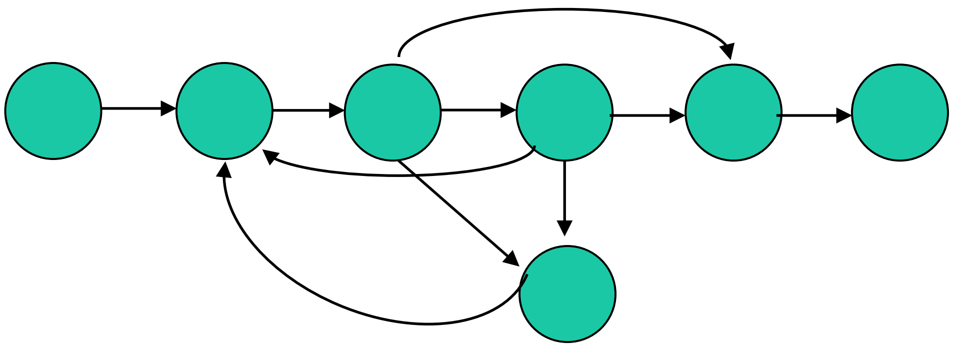
E= 1, N=2 , P=1
V= 1-2+2 = 1



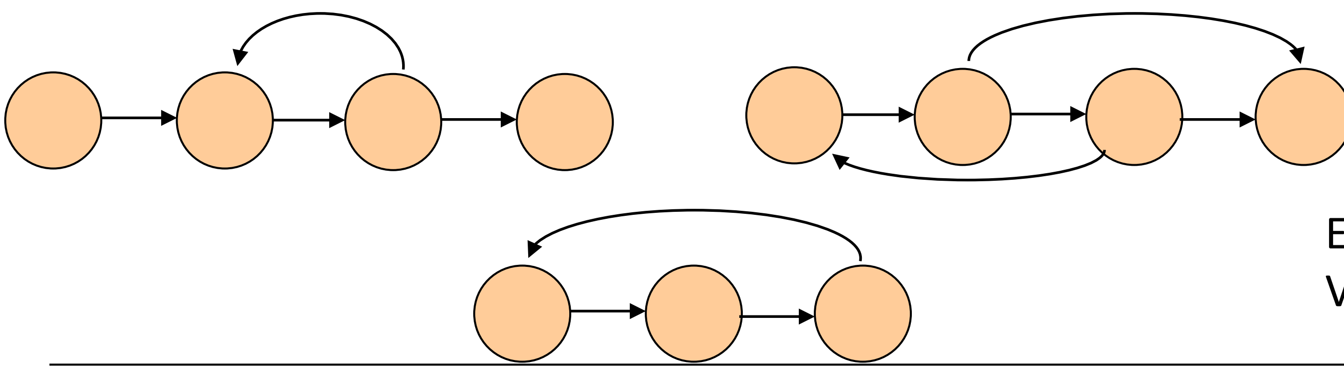
E= 4 , N=4 , P=1
V= 4-4+2 = 2



E= 2 , N= 4 , P=2
V= 2-4+4 = 2



E= 10, N= 7 , P=1
V= 10-7+2 = 5



E= 12, N= 11 , P= 3
V= 12-11+6 = 7

Agenda:

~~Terminology and scope~~

~~Quality factors revisited~~

~~Model of failure-based metrics~~

~~Usability metrics~~

~~Cyclomatic complexity~~

Examples of metrics and quality factors

Practical use

Software metrics

- Usage-based metrics
- Verification & Validation metrics
- Volume metrics
- Structural metrics
- Effort metrics
- Direct measurement
- Indirect measurement

Note: Pedagogical model only!

Usage based metrics - example

- Description: Number of good and bad features recalled by users.
- How to obtain data: Set up a test scenario. Let test users run the scenario. Collect number of good and bad features in a questionnaire afterwards.
- How to calculate the metric: Take the average of number of good and no. bad features. Two values.
- Relevant quality factor: Appropriateness recognizability – many good and few bad features indicates a good match with the users' mind-set.

Verification and validation metrics - example

- Description: Rate of severe defects found in inspection of design description.
- How to obtain data: Perform an inspection according to your process. Make sure that severity is in the classification scheme.
- How to calculate the metric: Divide the number of defects classified with highest severity with total number of defects in the Inspection record.
- Relevant quality factor: Functional correctness – a high proportion of severe defects in design indicates fundamental problems with the solution and/or competence.

Volume metrics - example

- Description: Number of non-comment lines of code.
- How to obtain data: Count non-comment lines of the code with a tool.
- How to calculate the metric: See above.
- Relevant quality factor: Reliability – it is often hard to understand a large portion of code, the fault density is often higher for large modules.

Structural metrics - example

- Description: Maximum depth of inheritance tree.
- How to obtain data: Count the depth of the inheritance tree for all classes with a tool.
- How to calculate the metric: Take the maximum value of the classes.
- Relevant quality factor: Analysability – It is hard to determine how a change in a higher class will affect inherited/overridden methods.

Effort metrics - example

- Description: Time spent in testing.
- How to obtain data: Make sure that testing activities are distinguished in time reporting forms. Make sure that all project activities are reported.
- How to calculate the metric: Sum the number of hours for all activities in testing for all people involved.
- Relevant quality factor: Testability – a comparably long testing time indicates low testability.

Agenda:

~~Terminology and scope~~

~~Quality factors revisited~~

~~Model of failure-based metrics~~

~~Usability metrics~~

~~Cyclomatic complexity~~

~~Examples of metrics and quality factors~~

Practical use

The Goal Question Metric approach

- Outside the course we can use a top-down approach: Goal-Question-Metric (GQM)

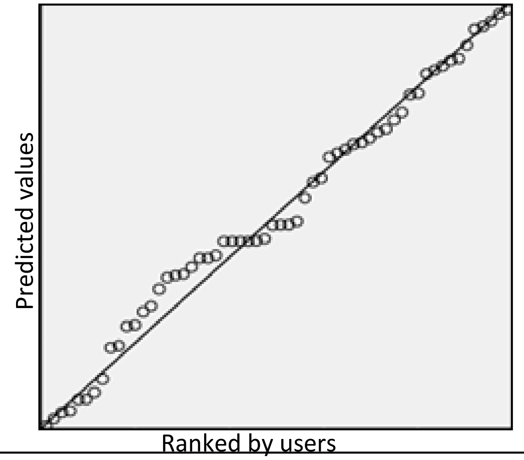
Goal	Purpose Issue Object (process) Viewpoint	Improve the timeliness of change request processing from the project manager's viewpoint
Question		What is the current change request processing speed?
Metrics		Average cycle time Standard deviation % cases outside of the upper limit
Question		Is the performance of the process improving?
Metrics		$\frac{\text{Current average cycle time}}{\text{Baseline average cycle time}} * 100$ Subjective rating of manager's satisfaction

Basili, Caldiera, Rombach (1994)

Research

Metric	Threshold Value	
Non-Self Transitions	-	
Transitions/State	Middle level state	Top Level State
	4 -5	3-4
State Depth	3	

Rank = 1.2 + 0.007NonSelfTransitions + 0.17Transitions/state + 0.25StateDepth



Rezaei, Ebersjö, Sandahl, Staron
Identifying and managing complex modules in executable software design models
IWSM Mensura 2014 conference

Summary

- Metrics and measurements are often confused
- Fault- and time based model
- Usability metrics
- Cyclomatic complexity
- Use the pedagogic model for inspiration

Software Metrics/Kristian Sandahl

www.liu.se

LINKÖPING UNIVERSITY